**丁丁书城网站详细设计说明书**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0. 导言**  0.1 目的  本文档的目的是描述淘宝进销存项目的详细设计，其主要内容包括：   * + 系统功能简介   + 系统详细设计简述   + 各个模块的分层结构   本文档的预期读者是：   * + 设计人员   + 开发人员   + 项目管理人员   + 测试人员   0.2 范围  该文档定义了系统的各个模块和模块接口，但未确定单元的具体实现，这部分内容将在编码实现中确定。  0.3 引用标准  [1] 《企业文档格式标准》V1.1， XXX软件有限公司。  [2] 《软件详细设计报告格式标准》V1.1，XXX软件有限公司软件工程过程化组织  0.4 参考资料  0.5 版本更新信息  本文档版本更新记录如表C-1所示：  表C-1 版本更新记录   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **修改编号** | **修改日期** | **修改后版本** | **修改位置** | **修改内容概述** | | 000 | 2016-6-3 | 0.1 | 全部 | 初始发布版本 | |  |  |  |  |  |   **1. 系统设计概述**  本系统主要针对管理员和顾客使用，管理员登录后台系统后可以对商品，订单等进行管理。其功能模块如下图所示：    **2. 详细设计概述**  本系统采用了jsp+servlet设计，在整个开发过程中，尽可能采用复用的原则，例如使用工厂模式，JAVA反射机制达到统一数据库操作，统一结果显示等。    **3. 详细设计**  **3.1用户注册模块**  用户注册模块负责前台用户成为网站会员，注册成功的用户信息添加到customer表中  3.1.1 实现流程图    3.1.2 输入输出规则  用户注册模块输入输出规则   |  |  | | --- | --- | | 模块名称 | 用户注册 | | 功能描述 | 用户注册成功后允许登录系统并享有注册用户的权限 | | 用户角色 | 游客 | | 模块触发条件 | 打开首页选择“立即注册” | | 输入规则 | 输入用户基本信息。要求符合格式，另外账号不能重复，邮箱不能重复 | | 输出结果 | 注册成功后打开登录界面 | | 关键算法 | 1.查询数据中是否存在相同的账号和邮箱  2.格式验证  3.验证码的生成和验证 | | 数据库操作 | 查询用户表，增加用户信息 |   3.1.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | register.jsp  login.jsp | CustomerServlet.java  LoginServlet.java | CustomerBiz.java  CustomerBizImpl.java | CustomerDao.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 注册界面 | register.jsp | 注册页面 | | 登陆页面 | login.jsp | 登录页面 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | UserAction | register.jsp | execute() () | 成功：login.jsp  失败：register.jsp | | ValidateCodeServlet | register.jsp | doPost() | 输出验证码 |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IUserBiz.java | public void saveUser(Users user); | 接收用户对象，将用户对象信息保存到数据库 | | UserBizImpl.java | public void saveUser(Users user) | 写具体的HQL语句，调用Dao层方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IUserDao.java | IGenericDao | 基类接口应用泛型和反射机制，对数据的操作进行统一处理，无需各持久层接口设计方法。 | | UserDaoImpl | IUserDao  AGenericHibernateDao | UserDaoImpl继承AgenericHibernateDao类，实现IUserDao接口 |   3.1.4 功能实现时序图    **3.2 用户登录模块**  用户登录是针对已注册的用户，登录成功后，注册用户获得购物交易的相应权限。  3.2.1 实现流程图    3.2.2 输入输出规则  登录模块输入输出规则   |  |  | | --- | --- | | 模块名称 | 登录 | | 功能描述 | 用户登录系统以达到注册用户的系统权限 | | 用户角色 | 注册用户 | | 模块触发条件 | 直接选择登录或在将商品加入购物车时转向登录 | | 输入规则 | 按照注册时输入的用户名和密码及随机验证码。可以选择两周内自动登录，则用户名和密码将存入cookie;如果不选将用户对象存入session | | 输出结果 | 登录成功后打开首页，登录用户信息存放在cookie中或者是session中，用户可以进行购买商品或对自己的信息进行维护。 | | 关键算法 | 1.查询数据库中用户名和密码  2.验证码的生成和验证  3.两周内自动登录的选择  4.根据用户ID获得用户默认收货地址 | | 数据库操作 | 查询用户表 |   3.2.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | login.jsp  index.jsp  header.jsp  footer.jsp | UserAction.java  ValidateCodeServlet.java | IUserBiz.java  UserBizImpl.java  IReceiverBiz.java  RecieverBizImpl.java | IUserDao.java  UserDaoImpl.java  IReceiverDao.java  RecieverDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 登录界面 | login.jsp | 登录主页面 | | 首页 | index.jsp | 前台首页面，包括header.jsp和footer.jsp,是为了网站统一风格 | | 页面上端 | header.jsp | 复用页面，记载用户登录信息及个人信息维护菜单以及网站Logo等信息 | | 页面尾端 | footer.jsp | 复用页面，显示广告及友情链接信息 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | UserAction | login.jsp | UserLogin() | 成功：index.jsp  失败：login.jsp | | ValidateCodeServlet | login.jsp | doPost() | 输出验证码 |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IUserBiz.java | public Users CheckLogin(final String username,final String password); | 接收用户名和密码查询数据库返回Users对象 | | UserBizImpl.java | public Users CheckLogin(final String username,final String password); | 写具体的HQL语句，调用数据持久层方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IUserDao.java | IGenericDao | 基类接口应用泛型和反射机制，对数据的操作进行统一处理，无需各持久层接口设计方法。 | | UserDaoImpl.java | IUserDao  AGenericHibernateDao | UserDaoImpl继承AgenericHibernateDao类，实现IUserDao接口 |   3.2.4 功能实现时序图    **3.3 商品搜索**  用户可以根据网站列出的图书品牌、款式进行快速搜索图书。还可以根据商品名字进行模糊搜素。搜索到的图书以列表形式分页显示，可以查看某本书的详细信息。  本环节首先说明品牌搜索实现过程  3.3.1 种类搜索实现流程图      **3.2 商品搜索-根据商品名和产地模糊检索**  3.2.1 模糊搜索实现流程图    3.5.2 模糊搜索输入输出规则  图书模糊搜索显示输入输出规则   |  |  | | --- | --- | | 模块名称 | 图书名称模糊搜索显示 | | 功能描述 | 根据图书名称搜索图书 | | 用户角色 | 注册用户或游客 | | 模块触发条件 | 打开网站首页 | | 输入规则 | 输入图书名称 | | 输出结果 | 显示包含名称关键字的所有图书列表 | | 关键算法 | 1. 获取表单图书名称  2. Action中获取关键字  3. 业务层根据关键字查询bookinfo表 | | 数据库操作 | 查询Book表 |   3.5.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | header.jsp search.jsp | BookServlet.java | BookBiz.java  BookBizImpl.java | BookDao.java  BookDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 网站头部独立页面 | header.jsp | 接收关键字和性别选择 | | 搜索结果页面 | Search.jsp,其中包含search\_result.jsp | 根据前置条件将搜索结果列表显示 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | BookAction | header.jsp | FuzzySearch() | search.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IBookBiz.java | public PageBean FuzzySearch(Object[] params,final int currentPage,final int pageSize); | 根据图书名称，产地关键字及性别，当前页数和每页记录数查询数据库中图书信息返回分页信息。 | | BookBizImpl.java | public PageBean FuzzySearch(Object[] params,final int currentPage,final int pageSize); | 写具体的HQL语句，调用数据持久层findByPageBean()方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IShoeDao.java | IGenericDao | 获取分页信息的统一方法  public PageBean findByPageBean(final String hql,final Object[] param,final int currentpage,final int pageSize); | | BookDaoImpl.java | IBookDao  AGenericHibernateDao | BookDaoImpl继承AgenericHibernateDao类，实现IBookDao接口 |   3.5.4 功能实现时序图（同3.3.4）  **3.7 商品详情**  3.7.1 商品详情实现流程图    3.7.2 商品详情输入输出规则  商品详情输入输出规则   |  |  | | --- | --- | | 模块名称 | 商品详情显示 | | 功能描述 | 显示指定图书的详细信息 | | 用户角色 | 注册用户或游客 | | 模块触发条件 | 打开商品列表页 | | 输入规则 | 点击图书图片或商品名称 | | 输出结果 | 显示指定图书的详细信息 | | 关键算法 | 1. 图书列表中具有图书sid参数值  2. Action中获取sid值  3. 在业务层中通过sid值获取图书的详细信息 | | 数据库操作 | 查询Book表 |   3.7.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | search\_result.jsp  product\_detail.jsp | BookAction.java | IBookBiz.java  BookBizImpl.java | IBookDao.java  BookDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 图书列表页面 | search\_result.jsp | 显示图书列表 | | 图书详情页面 | product\_detail.jsp | 显示图书的详细信息 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | BookAction | Search\_result.jsp | FindBySid() | product\_detail.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IBookBiz.java | public Book FindById(final int sid); | 根据图书id获取图书对象 | | BookBizImpl.java | public Book FindById(final int sid); | 写具体的HQL语句，调用数据持久层findById(sid)方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IShoeDao.java | IGenericDao | 获取分页信息的统一方法  public T findById(ID id); | | BookDaoImpl.java | IBookDao  AGenericHibernateDao | BookDaoImpl继承AgenericHibernateDao类，实现IBookDao接口 |   3.7.4 功能实现时序图    **3.8 添加购物车**  3.8.1 添加购物车实现流程图    3.8.2 添加购物车输入输出规则  添加购物车输入输出规则   |  |  | | --- | --- | | 模块名称 | 添加购物车 | | 功能描述 | 将图书添加到购物车 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户打开商品详情页 | | 输入规则 | 选择尺码后添加到购物车 | | 输出结果 | 打开购物车页面，显示购物车信息 | | 关键算法 | 1.页面判断是否选择尺码和个人是否有默认收货地址  2.获取图书编号，尺码和数量(默认为1)  3. 添加到购物车的Action中根据图书id,尺码，购买数量组合Bookizes对象 ,Bookizes表中包含各图书对应尺寸的库存。  4. 如果图书没有被删除，并且库存量大于购买量，将图书信息添加到Map集合（购物车 Cart ）,并添加到session中 | | 数据库操作 | 查询Bookizes表 |   3.8.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | product\_detail.jsp  cart.jsp  cart\_table.jsp | CartAction.java | IBookizesBiz.java  BookizesBizImpl.java  IReceiverBiz.java  ReceiverBizImpl.java | IBookizesDao.java  BookizesDaoImpl.java  IReceiverDao.java  ReceiverDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 图书详情页面 | product\_detail.jsp | 显示图书的详细信息 | | 购物车页面 | cart.jsp,其中包括cart\_table.jsp | 显示购物车的详细信息 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | CartAction | product\_detail.jsp | execute() | cart.jsp | | CartAction | Product\_detail.jsp | getDefaultAddress() | personal\_center.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IBookizesBiz.java | public Bookizes FindByBookid(final int sid,final int count,final float sizes); | 根据图书id,数量，尺寸获取图书尺寸信息对象（包含库存） | | BookizesBizImpl.java | public Bookizes FindByBookid(final int sid,final int count,final float sizes); | 写具体的HQL语句，调用数据持久层findByObject方法 | | IReceiverBiz.java | public Receives FindByDefault(int uid); | 根据用户id获得默认收货地址 | | ReceiverBizImpl.java | public Receives FindByDefault(int uid); | 写具体的HQL语句，调用数据持久层findByObject方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IBookizesDao.java | IGenericDao | 根据条件参数获取集合  public List<T> findByObject(String hql,Object[] param); | | BookizesDaoImpl.java | IBookizesDao  AGenericHibernateDao | BookizesDaoImpl  继承AgenericHibernateDao类，实现IBookizesDao接口 | | IReceiverDao.java | IGenericDao | 根据条件参数获取集合  public List<T> findByObject(String hql,Object[] param); | | ReceiverDaoImpl.java | IReceiverDao AGenericHibernateDao | ReceiverDaoImpl  继承AgenericHibernateDao类，实现IReceiverDao接口 |   3.8.4 功能实现时序图    **3.9 删除购物车商品**  3.9.1 删除购物车商品实现流程图    3.9.2删除购物车商品输入输出规则  删除购物车商品输入输出规则   |  |  | | --- | --- | | 模块名称 | 删除购物车商品 | | 功能描述 | 将图书从购物车中删除 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户在购物车页面选择删除商品 | | 输入规则 | 点击“删除商品”链接 | | 输出结果 | 显示删除商品后的购物车信息页面 | | 关键算法 | 1. 获取页面隐藏域信息  2. 判断当前购物车中有几种商品（根据单个商品总价）  3. 如果存在多种商品，通过js操作DOM对象删除商品  4. 如果只剩下一种商品，则将购物车清空，实际操作是将购物车对象从session中删除。 | | 数据库操作 | 无 |   3.9.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | cart.jsp  cart\_table.jsp | OrderAction.java | 无 | 无 |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 购物车页面 | cart.jsp,其中包括cart\_table.jsp | 显示购物车的详细信息及相关js操作 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | OrderAction | cart\_table.jsp | deleteCart() | index.jsp,清空购物车后返回首页 |   3.9.4 功能实现时序图  不清空购物车（略）  清空购物车    **3.10 结算购物车商品**  3.10.1 结算购物车商品实现流程图    3.10.2结算购物车商品输入输出规则  结算购物车商品输入输出规则   |  |  | | --- | --- | | 模块名称 | 结算购物车商品 | | 功能描述 | 确定购物车中商品，提交结算 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户在购物车页面选择“去结算” | | 输入规则 | 点击“去结算”链接 | | 输出结果 | 显示购物车所有商品结算信息和个人默认收货地址信息 | | 关键算法 | 1. 获取购物车信息  2. 获取个人默认收货地址信息 | | 数据库操作 | 查询receives |   3.10.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | cart.jsp  cart\_table.jsp | OrderAction.java | IReceiverBiz.java  ReceiverBizImpl.java | IReceiverDao.java  ReceiverDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 购物车页面 | cart.jsp,其中包括cart\_table.jsp | 显示购物车的详细信息及相关js操作 | | 订单页面 | ordersheet.jsp | 订单信息页面（购物车和收货地址） |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | OrderAction | cart\_table.jsp | execute() | ordersheet.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IReceiverBiz.java | public Receives FindByDefault(int uid); | 根据用户id获得默认收货地址 | | ReceiverBizImpl.java | public Receives FindByDefault(int uid); | 写具体的HQL语句，调用数据持久层findByObject方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IReceiverDao.java | IGenericDao | 根据条件参数获取集合  public List<T> findByObject(String hql,Object[] param); | | ReceiverDaoImpl.java | IReceiverDao AGenericHibernateDao | ReceiverDaoImpl  继承AgenericHibernateDao类，实现IReceiverDao接口 |   3.10.4 功能实现时序图    **3.11直接购买商品**  3.11.1 直接购买商品实现流程图    3.11.2直接购买商品输入输出规则   |  |  | | --- | --- | | 模块名称 | 直接购买商品 | | 功能描述 | 在商品详情页选择直接购买商品，功能类似添加到购物车，只是直接购买的商品不经过购物车页面，直接到订单页 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户在详情页选择“直接购买” | | 输入规则 | 选择尺码后点击“直接购买”链接 | | 输出结果 | 打开订单页，显示购物车所有商品结算信息和个人默认收货地址信息 | | 关键算法 | 1. 将商品添加到购物车但不现实购物车页面  2. 获取个人默认收货地址信息 | | 数据库操作 | 查询Bookizes表和receives表 |   3.11.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | cart.jsp  cart\_table.jsp | CartAction.java | IBookizesBiz.java  BookizesBizImpl.java  IReceiverBiz.java  ReceiverBizImpl.java | IBookizesDao.java  BookizesDaoImpl.java  IReceiverDao.java  ReceiverDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 商品详情页面 | product\_detail.jsp | 显示商品详情及“直接购买”链接 | | 订单页面 | ordersheet.jsp | 订单信息页面（购物车和收货地址） |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | CartAction | product\_detail.jsp | Buynow() | ordersheet.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IReceiverBiz.java | public Receives FindByDefault(int uid); | 根据用户id获得默认收货地址 | | ReceiverBizImpl.java | public Receives FindByDefault(int uid); | 写具体的HQL语句，调用数据持久层findByObject方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | IReceiverDao.java | IGenericDao | 获取分页对象方法findByPageBean | | ReceiverDaoImpl.java | IReceiverDao AGenericHibernateDao | ReceiverDaoImpl继承AgenericHibernateDao类，实现IReceiverDao接口 |   3.11.4 功能实现时序图    **3.12秒杀商品**  3.12.1 秒杀商品实现流程图    3.12.2秒杀商品输入输出规则   |  |  | | --- | --- | | 模块名称 | 秒杀商品 | | 功能描述 | 有些商品在特定时期内价格优惠，用户可以选择抢在这期间提交订单，俗成“秒杀” | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户在秒杀页面“秒杀” | | 输入规则 | 点击“秒杀”链接 | | 输出结果 | 打开订单页，显示秒杀商品结算信息和个人默认收货地址信息 | | 关键算法 | 1. 列出所有处于秒杀期间的商品  2. 判断用户的积分和商品秒杀所需积分  3. 获取个人默认收货地址 | | 数据库操作 | 查询Bookizes表，receives表，second\_kills表 |   3.12.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | index.jsp  secondKillsLst.jsp secondKills.jsp  ordersheet.jsp | SecondKillsAction.java  OrderAction.java | ISecondKillsBiz.java  SecondKillsBizImpl.java  IReceiverBiz.java  ReceiverBizImpl.java | ISecondKillsDao.java  SecondKillsDaoImpl.java  IReceiverDao.java  ReceiverDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 首页 | index.jsp | 显示“秒杀”链接，转向秒杀商品页面 | | 秒杀有效商品显示页面 | secondKillsLst.jsp | 有效的，可以进行秒杀的商品列表，分页显示 | | 秒杀执行页面 | secondKills.jsp | 通过秒杀商品列表选择某个商品，打开该商品的详细信息和执行秒杀操作 | | 订单页面 | ordersheet.jsp | 执行秒杀后转向订单页 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | SecondKillsAction.java | index.jsp | execute() | secondKillsLst.jsp | | SecondKillsAction.java | secondKillsLst.jsp | FindBySkid() | secondKills.jsp | | OrderAction.java | secondKills.jsp | addSecondKill() | ordersheet.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | ISecondKillsBiz.java | //获取有效秒杀商品  public PageBean findAvaliable(int currentpage,int pagesize);  //获取单个秒杀商品对象  public SecondKills FidnById(int skid); | 获取有效秒杀商品在secondKillsLst.jsp中显示，获取单个秒杀商品对象是在secondKills.jsp中显示 | | SecondKillsBizImpl.java | //获取有效秒杀商品  public PageBean findAvaliable(int currentpage,int pagesize);  //获取单个秒杀商品对象  public SecondKills FidnById(int skid); | 写具体的HQL语句访问持久层findByPageBean和findByObject方法 | | IReceiverBiz.java | public Receives FindByDefault(int uid); | 根据用户id获得默认收货地址 | | ReceiverBizImpl.java | public Receives FindByDefault(int uid); | 写具体的HQL语句，调用数据持久层findByObject方法 |   数据持久层设计   |  |  |  | | --- | --- | --- | | **持久层接口** | **基类接口** | **接口说明** | | ISecondKillsDao.java | IGenericDao | 1.获取分页对象方法findByPageBean  2.根据条件获取单个对象方法findByObject | | SecondKillsDaoImpl.java | ISecondKillsDao  AGenericHibernateDao | SecondKillsDaoImpl继承AgenericHibernateDao类，实现ISecondKillsDao接口 | | IReceiverDao.java | IGenericDao | 获取默认对象方法findByObject | | ReceiverDaoImpl.java | IReceiverDao AGenericHibernateDao | ReceiverDaoImpl继承AgenericHibernateDao类，实现IReceiverDao接口 |   3.12.4 功能实现时序图  1）获取秒杀商品列表    2）打开秒杀商品详情页    3）执行秒杀    **3.14生成订单**  3.14.1 生成订单实现流程图    3.14.2生成订单输入输出规则   |  |  | | --- | --- | | 模块名称 | 生成订单 | | 功能描述 | 购买商品后转向订单页，在订单页输入用户备注信息提交订单，实际应用中需要支付成功后才修改数据库数据，本项目为演示效果省略支付部分，用户提交订单后就修改数据库 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户购买商品后在订单页“提交订单” | | 输入规则 | 输入用户备注信息 | | 输出结果 | 修改数据库商品库存，生成订单信息 | | 关键算法 | 1.获取订单页商品信息  2.获取登录用户信息  3.修改商品库存，生成订单信息，修改商品售出次数 | | 数据库操作 | 查询spcify\_Book表，receives表 |   3.14.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | ordersheet.jsp  alipay.jsp | OrderAction.java | IOrdersBiz.java  IBookizesBiz.java  IUsersBiz.java  IOrderBookBiz.java  ISecondKillsBiz.java  IBookBiz.java  ISpcifyResultsBiz.java | IOrdersDao.java  IBookizesDao.java  IUsersDao.java  IOrderBookDao.java  ISecondKillsDao.java  IBookDao.java  ISpcifyResultsDao.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 订单页面 | ordersheet.jsp | 执行定制后转向订单页 | | 支付页面 | alipay.jsp | 支付接口页面 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | OrderAction.java | ordersheet.jsp | createOrder() | alipay.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IOrdersBiz.java | public void createOrder(Orders order); | 生成订单 | | IBookizesBiz.java | public void ModifyShoeCount(int ssid,int count); | 更新相应尺码图书数量 | | IUsersBiz.java | public abstract void UpdateUser(Users user); | 更新用户信息，主要是更新用户积分 | | IOrderBookBiz.java | public void createOrderBook(OrderBook orderBook); | 生成订单详单 | | ISecondKillsBiz.java | public void UpdateSecondKills(SecondKills kill); | 更新一个秒杀商品 | | IBookBiz.java | public void ModifyTimessold(int sid,int count) ; | 修改图书售出件数（新增） | | ISpcifyResultsBiz.java | public void CreateSpcifyBook(SpcifyResults spcify); | 添加定制鞋订单 |   数据持久层设计(略)  3.14.4 功能实现时序图    **3.15 个人信息管理-收货地址**  3.15.1 收货地址管理实现流程图    3.15.2收货地址管理输入输出规则   |  |  | | --- | --- | | 模块名称 | 收货地址管理 | | 功能描述 | 用户购买商品提交订单时，需向系统提供默认收货地址，这样商家才可以根据收货地址给用户发货，该模块就是对注册用户收货地址的增删改查操作。 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户进入个人中心 | | 输入规则 | 基本资料增删改查 | | 输出结果 | 更新用户对应的收货地址信息 | | 关键算法 | 1.获取表单值  2.更新收货地址信息 | | 数据库操作 | 操作receives表 |   3.15.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | personal\_center.jsp  personal\_center\_menu.jsp  personal\_center\_rcvAdd.jsp | ReceiversAction.java | IReceiverBiz.java  ReceiverBizImpl.java  IUsersBiz.java  UsersBizImpl.java | IReceiverDao.java  ReceiverDaoImpl.java  IUsersDao.java  UsersDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 个人中心首页 | personal\_center.jsp | 打开个人中心，显示个人信息及订单信息 | | 个人中心菜单页 | personal\_center\_menu.jsp | 提供个人中心管理菜单项 | | 收货地址管理页 | personal\_center\_rcvAdd.jsp | 管理个人收货地址 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | ReceiversAction.java | personal\_center\_rcvAdd.jsp | AddReceive()  DeleteReceive()  ModifyReceive()  MainReceive()  execute() | personal\_center\_rcvAdd.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IReceiverBiz.java | //分页获取地址信息  public PageBean SearchReceiversByuid(int uid,int currentPage,int pageSize);  //判断默认地址唯一性  public boolean IsAddress();  //添加收货地址信息  public void SaveReceives(Receives receive);  //获取单个地址对象  public Receives FindByrecid(int recid);  //修改地址信息  public void ModifyReceive(Receives receive);  //删除地址信息  public void DeleteReceive(int recid);  //设置默认地址  public boolean SetDefaultAddress(int recid);  //获取默认地址  public Receives FindByDefault(int uid); | 关于收货地址的增删改查操作 | | ReceiverBizImpl.java | 略 | 实现IReceiverBiz接口，写具体的HQL语句调用Dao层方法 | | IUsersBiz.java | public abstract Users FindPwd(final String uaccount); | 根据账号获取用户对象 | | UsersBizImpl.java | 略 | 实现IUsersBiz接口，写具体的HQL语句调用Dao层方法 |   数据持久层设计(略)  3.15.4 功能实现时序图    **3.16 个人信息管理-个人信息修改**  3.16.1 个人信息修改实现流程图    3.16.2个人信息修改输入输出规则   |  |  | | --- | --- | | 模块名称 | 个人信息修改 | | 功能描述 | 注册为网站的用户登录后可以对自己的个人信息进行修改，包括个人基本信息修改和密码修改 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户进入个人中心 | | 输入规则 | 个人信息修改 | | 输出结果 | 更新用户对应的基本信息和登录密码 | | 关键算法 | 1.获取表单值  2.更新个人信息 | | 数据库操作 | 操作users表 |   3.16.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | personal\_center.jsp  personal\_center\_menu.jsp  personal\_center\_modify\_pwd.jsp | UsersAction.java | IUsersBiz.java  UsersBizImpl.java | IUsersDao.java  UsersDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 个人中心首页 | personal\_center.jsp | 打开个人中心，显示个人信息及订单信息 | | 个人中心菜单页 | personal\_center\_menu.jsp | 提供个人中心管理菜单项 | | 信息修改页面 | personal\_center\_modify\_pwd.jsp | 修改个人基本信息和密码 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | UsersAction.java | personal\_center\_modify\_pwd.jsp | UpdatePersonal()  UpdatePassword() | personal\_center\_user\_info.jsp  personal\_center\_modify\_pwd.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IUsersBiz.java | public abstract void UpdateUser(Users user); | 根据接收Users对象修改用户信息 | | UsersBizImpl.java | public abstract void UpdateUser(Users user); | 实现IUsersBiz接口，写具体的HQL语句调用Dao层方法 |   数据持久层设计(略)  3.16.4 功能实现时序图  1）修改个人信息    2）修改密码    **3.17 个人信息管理-个人订单管理**  3.17.1 个人订单管理实现流程图    3.17.2个人订单管理输入输出规则   |  |  | | --- | --- | | 模块名称 | 个人订单管理 | | 功能描述 | 用户购买商品后将提交订单，提交后由网站后台管理员处理订单。则用户可以在个人中心查看自己下单成功的订单、秒杀的订单、发货途中的订单、交易成功的订单、退货的订单等。 | | 用户角色 | 注册用户 | | 模块触发条件 | 登录用户进入个人中心 | | 输入规则 | 对订单信息的查看，取消订单，取消退货，确认收货 | | 输出结果 | 个人订单信息的修改 | | 关键算法 | 1.修改订单状态ostate  待发货订单（下单成功）ostate=1  发货途中的订单ostate=2  交易成功的订单ostate=3  申请退货中的订单ostate=4  已退货订单ostate=5  已取消订单ostate=6  秒杀的订单ostate=7 | | 数据库操作 | 操作orders表 |   3.17.3视图层、Action、业务层、数据持久层的说明   |  |  |  |  | | --- | --- | --- | --- | | **视图** | **Action/Servlet** | **业务层（biz）** | **数据持久层（dao）** | | order.jsp | OrderAction.java | IOrderBiz.java  OrderBizImpl.java | IOrderDao.java  OrderDaoImpl.java |   视图层页面设计   |  |  |  | | --- | --- | --- | | **界面** | **JSP** | **功能描述** | | 个人订单显示页面 | order.jsp | 打开个人中心，显示个人各种状态下的订单 |   Action/Servlet设计   |  |  |  |  | | --- | --- | --- | --- | | **Action/Servlet** | **入口** | **方法** | **出口** | | OrderAction.java | order.jsp | ShowOrder()  cancleOrder()  cancleBatch()  confirm()  TurnBackOrders() | order.jsp |   业务层设计   |  |  |  | | --- | --- | --- | | **业务接口** | **本模块业务方法** | **方法说明** | | IOrderBiz.java | //分页显示订单列表  public PageBean FindOrderByPage(int uid,int ostate,int currentpage,int pagesize);  //取消单个订单  public void cancleOrder(int oid);  //取消批量订单  public void BatchcancleOrder(String hql);  //获取物流信息  public Orders getExpInfo(int oid);  //更新订单状态  public boolean UpdateOrderState(int oid,int ostate); | 获取指定用户订单信息，取消订单，更新订单信息，获取物流信息 | | OrderBizImpl.java | 略 | 实现IOrderBiz接口，写具体的HQL语句调用Dao层方法 |   数据持久层设计(略)  3.17.4 功能实现时序图 |